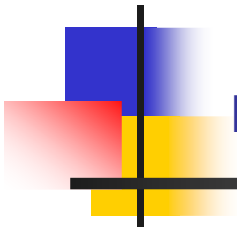
Reti di Calcolatori
AA 2009/2010

UNIVERSITÀ DEGLI STUDI DI TRENTO



http://disi.unitn.it/locigno/index.php/teaching-duties/computer-networks

Renato Lo Cigno

e

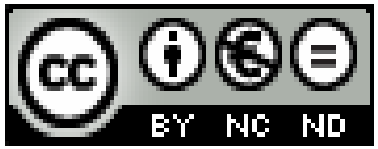
Claudio Covelli

Copyright

Quest'opera è protetta dalla licenza:

Creative Commons
Attribuzione-Non commerciale-Non opere derivate
2.5 Italia License

Per i dettagli, consultare
http://creativecommons.org/licenses/by-nc-nd/2.5/it/



Introduzione ...

Cos'è Internet?

Ai confini della rete

- sistemi terminali, reti di accesso, collegamenti

Il nucleo della rete

- commutazione di circuito e di pacchetto, struttura della rete

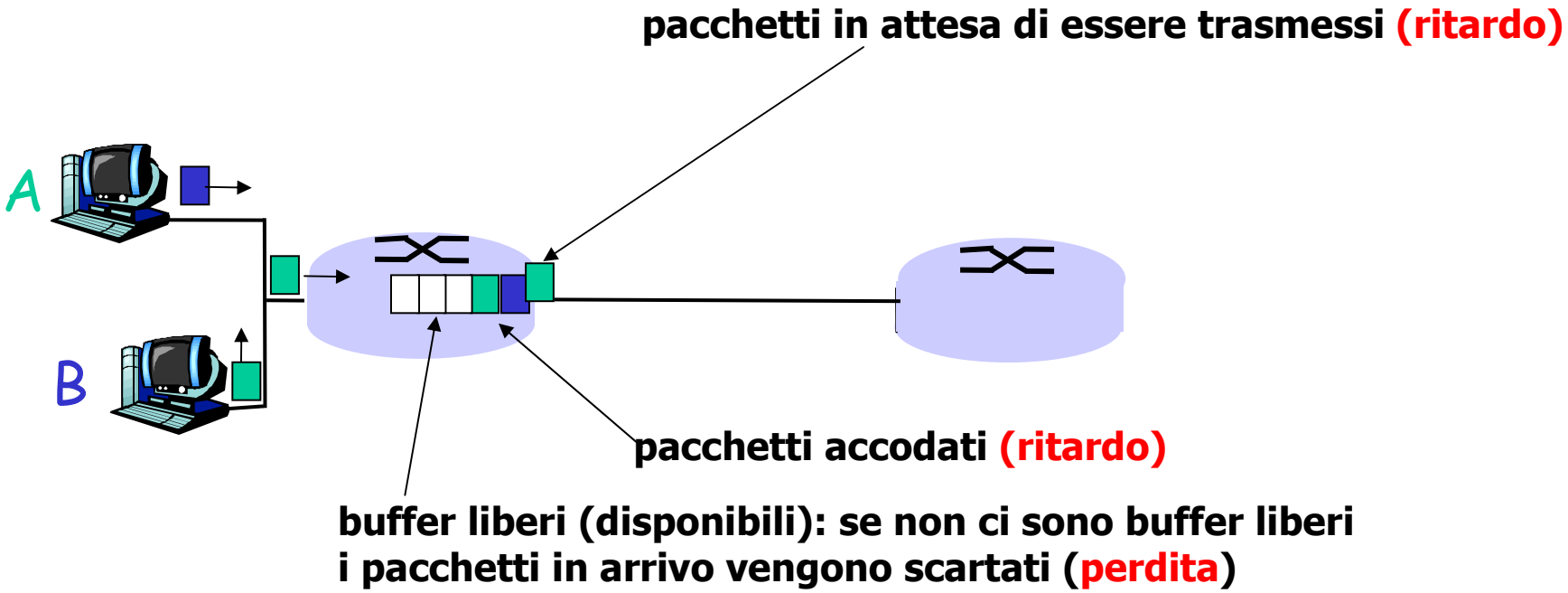
Ritardi, perdite e throughput nelle reti a commutazione di pacchetto

Livelli di protocollo e loro modelli di servizio

Reti sotto attacco: la sicurezza

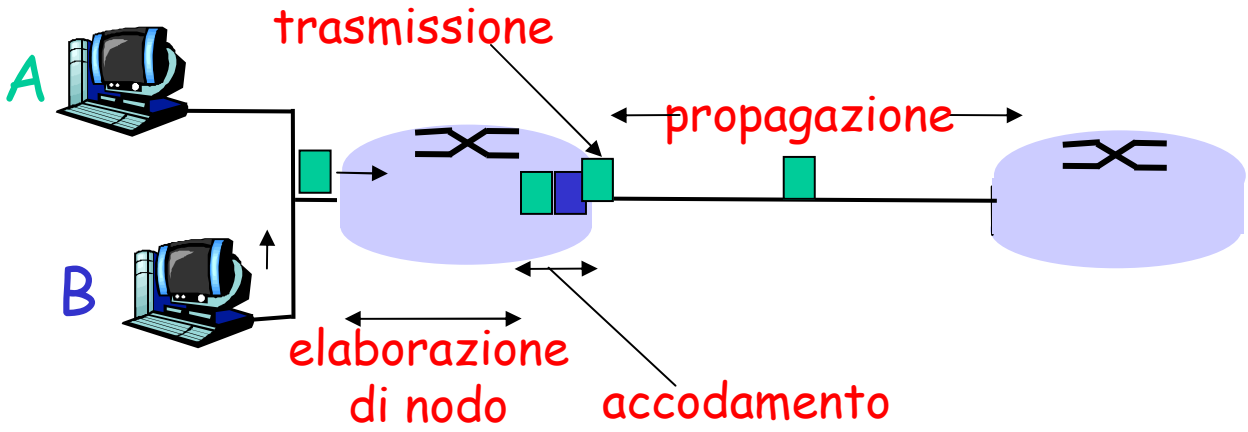
Come si verificano ritardi e perdite?

- I pacchetti *si accodano* nei buffer dei router
- il tasso di arrivo dei pacchetti sul collegamento eccede la capacità del collegamento di evaderli
- i pacchetti si accodano, in attesa del proprio turno



Quattro cause di ritardo per i pacchetti

- 1. Ritardo di elaborazione del nodo:
  - controllo errori sui bit
  - determinazione del canale di uscita
- 2. Ritardo di accodamento
  - attesa di trasmissione
  - livello di congestione del router



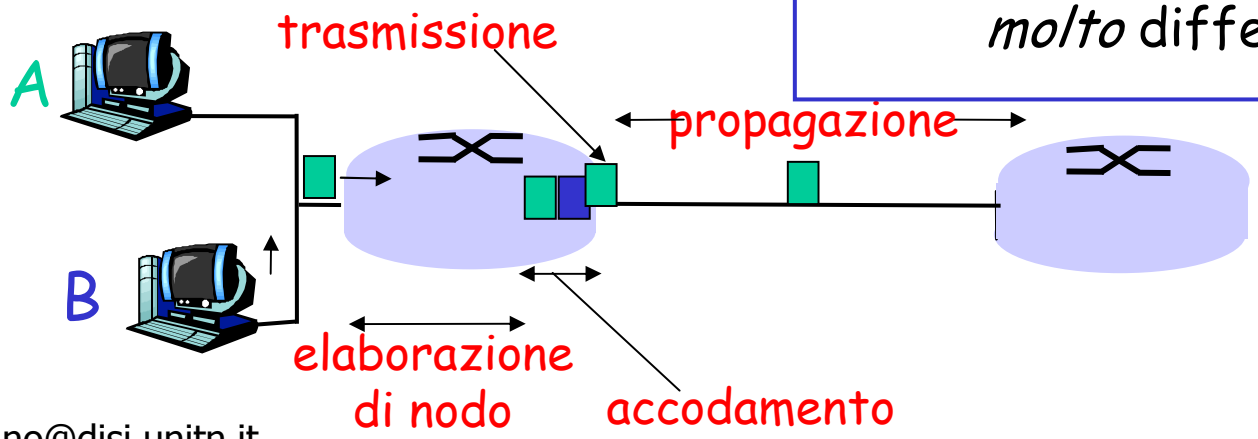
Ritardo nelle reti a commutazione di pacchetto

- 3. Ritardo di trasmissione (L/R):
  - R=frequenza di trasmissione del collegamento (in bit/s)
  - L=lunghezza del pacchetto (in bit)
  - Ritardo di trasmissione = L/R

- 4. Ritardo di propagazione (d/s)
  - d = lunghezza del collegamento fisico
  - s = velocità di propagazione del collegamento (~$2x10^8$ m/sec)
  - Ritardo di propagazione = d/s

Nota

s e R sono due quantità *molto* differenti!



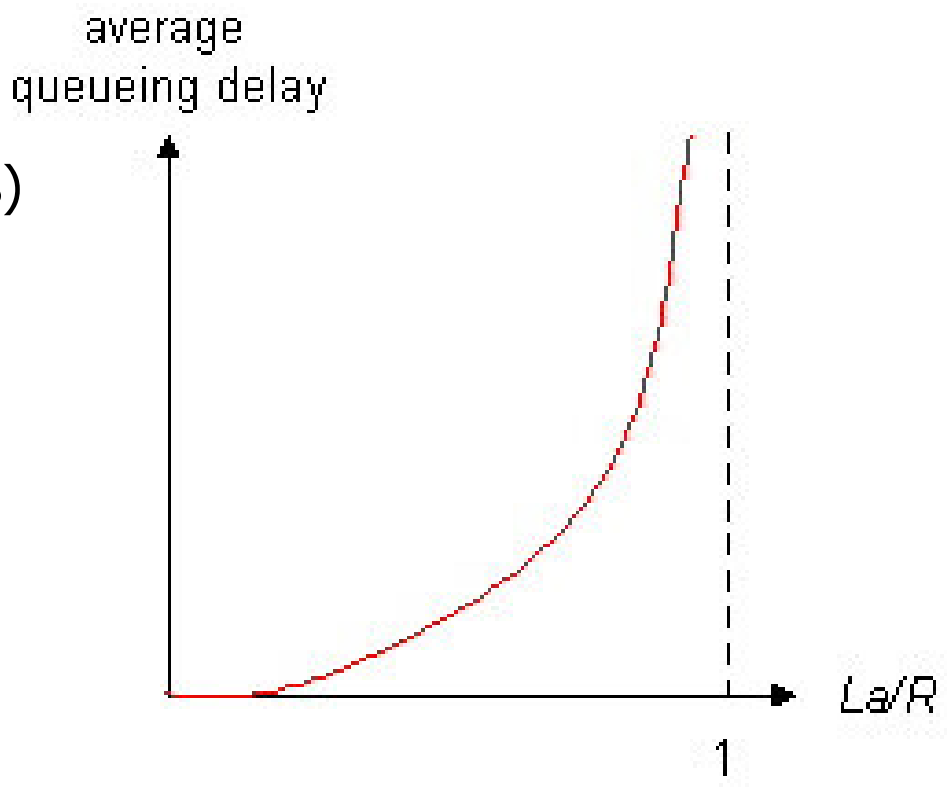
Ritardo di nodo

$$d_{\text{nodo}} = d_{\text{proc}} + d_{\text{queue}} + d_{\text{trans}} + d_{\text{prop}}$$

- d_{proc} = ritardo di elaborazione (*processing delay*)
  - in genere pochi microsecondi, o anche meno
- d_{queue} = ritardo di accodamento (*queuing delay*)
  - dipende dalla congestione
- d_{trans} = ritardo di trasmissione (*transmission delay*)
  - = L/R, significativo sui collegamenti a bassa velocità
- d_{prop} = ritardo di propagazione (*propagation delay*)
  - da pochi microsecondi a centinaia di millisecondi

Ritardo di accodamento

- R=frequenza di trasmissione (bps)
- L=lunghezza del pacchetto (bit)
- a=tasso medio di arrivo dei pacchetti

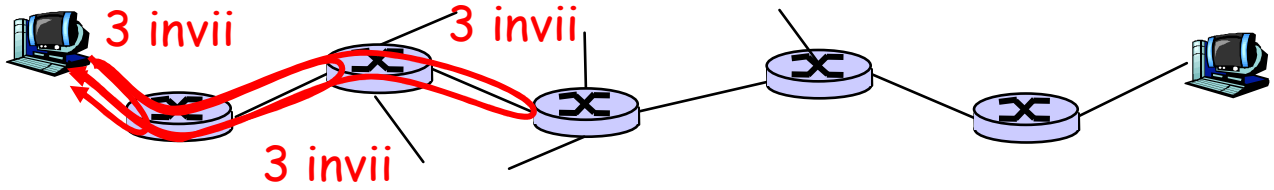


La/R = intensità di traffico

- La/R ~ 0: poco ritardo
- La/R -> 1: il ritardo si fa consistente
- La/R > 1: più "lavoro" in arrivo di quanto possa essere effettivamente svolto, ritardo medio infinito!

Ritardi e percorsi in Internet

- Ma cosa significano effettivamente ritardi e perdite nella "vera" Internet?
- **`Traceroute:`** programma diagnostico che fornisce una misura del ritardo dalla sorgente al router lungo i percorsi Internet punto-punto verso la destinazione.
  - invia tre pacchetti che raggiungeranno il router *i* sul percorso verso la destinazione
  - il router *i* restituirà i pacchetti al mittente
  - il mittente calcola l'intervallo tra trasmissione e risposta



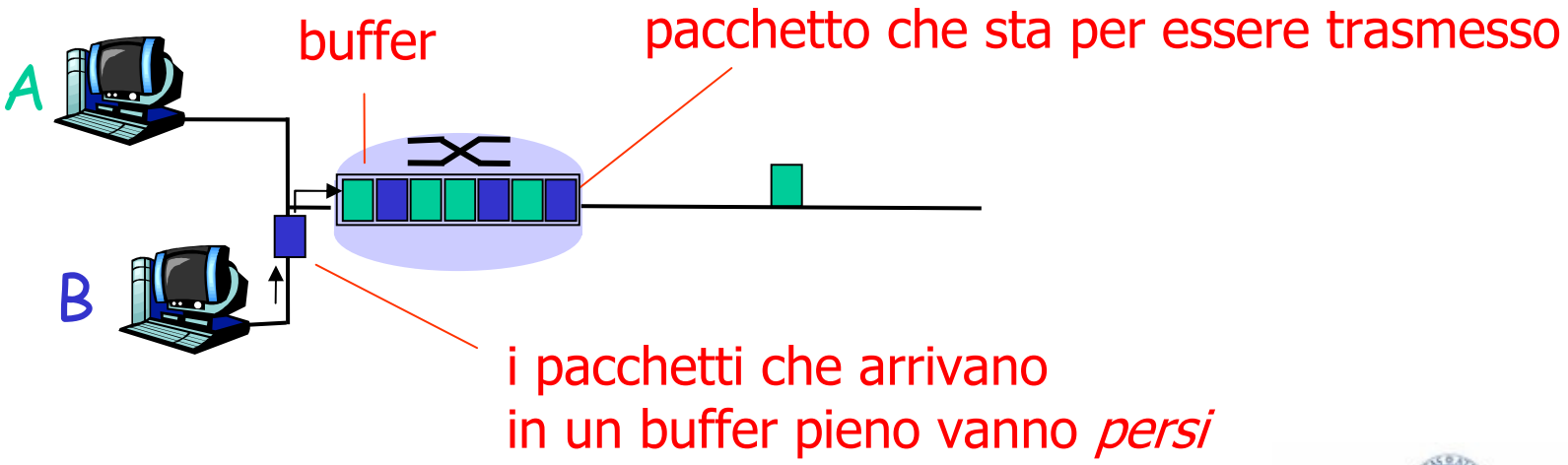
Ritardi e percorsi in Internet

traceroute: da gaia.cs.umass.edu a www.eurecom.fr


```
1  cs-gw (128.119.240.254)  1 ms  1 ms  2 ms
2  border1-rt-fa5-1-0.gw.umass.edu (128.119.3.145)  1 ms  1 ms  2 ms
3  cht-vbns.gw.umass.edu (128.119.3.130)  6 ms 5 ms 5 ms
4  jn1-at1-0-0-19.wor.vbns.net (204.147.132.129)  16 ms 11 ms 13 ms
5  jn1-so7-0-0-0.wae.vbns.net (204.147.136.136)  21 ms 18 ms 18 ms
6  abilene-vbns.abilene.ucaid.edu (198.32.11.9)  22 ms  18 ms  22 ms
7  nycm-wash.abilene.ucaid.edu (198.32.8.46)  22 ms  22 ms  22 ms
8  62.40.103.253 (62.40.103.253)  104 ms 109 ms 106 ms
9  de2-1.de1.de.geant.net (62.40.96.129)  109 ms 102 ms 104 ms
10  de.fr1.fr.geant.net (62.40.96.50)  113 ms 121 ms 114 ms
11  renater-gw.fr1.fr.geant.net (62.40.103.54)  112 ms  114 ms  112 ms
12  nio-n2.cssi.renater.fr (193.51.206.13)  111 ms  114 ms  116 ms
13  nice.cssi.renater.fr (195.220.98.102)  123 ms  125 ms  124 ms
14  r3t2-nice.cssi.renater.fr (195.220.98.110)  126 ms  126 ms  124 ms
15  eurecom-valbonne.r3t2.ft.net (193.48.50.54)  135 ms  128 ms  133 ms
16  194.214.211.25 (194.214.211.25)  126 ms  128 ms  126 ms
17  * * *
18  * * *
19  fantasia.eurecom.fr (193.55.113.142)  132 ms  128 ms  136 ms
```

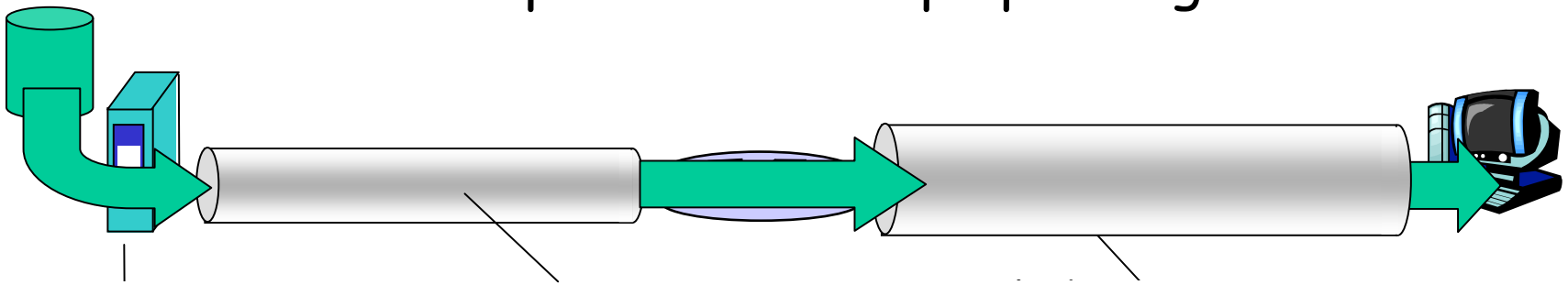
Perdita di pacchetti

- una coda (detta anche buffer) ha capacità finita
- quando il pacchetto trova la coda piena, viene scartato (e quindi va perso)
- il pacchetto perso può essere ritrasmesso dal nodo precedente, dal sistema terminale che lo ha generato, o non essere ritrasmesso affatto



Throughput

- *throughput:* frequenza (bit/unità di tempo) alla quale i bit sono trasferiti tra mittente e ricevente
  - *instantaneo:* in un determinato istante
  - *medio:* in un periodo di tempo più lungo



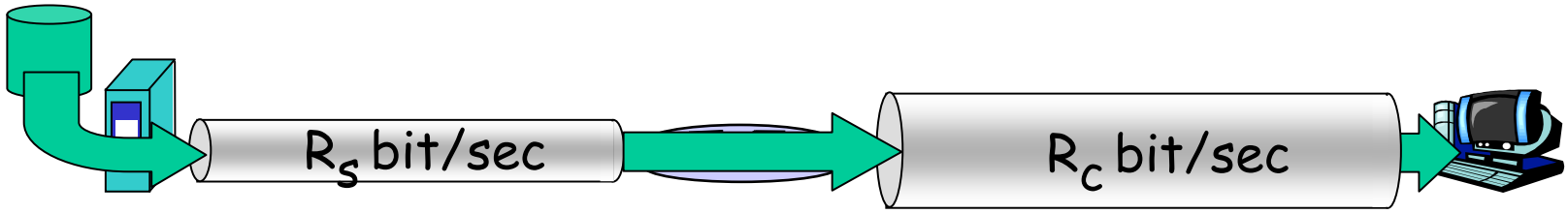
il server invia bit (fluido) nel tubo

tubo che può trasportare fluido a R_s bit/sec

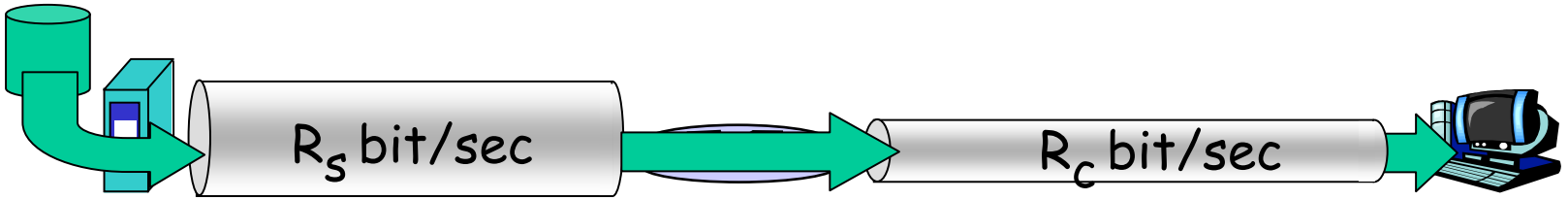
tubo che può trasportare fluido a R_c bits/sec

Throughput (segue)

- $R_s < R_c$ Qual è il throughput medio end to end?



- $R_s > R_c$ Qual è il throughput medio end to end?

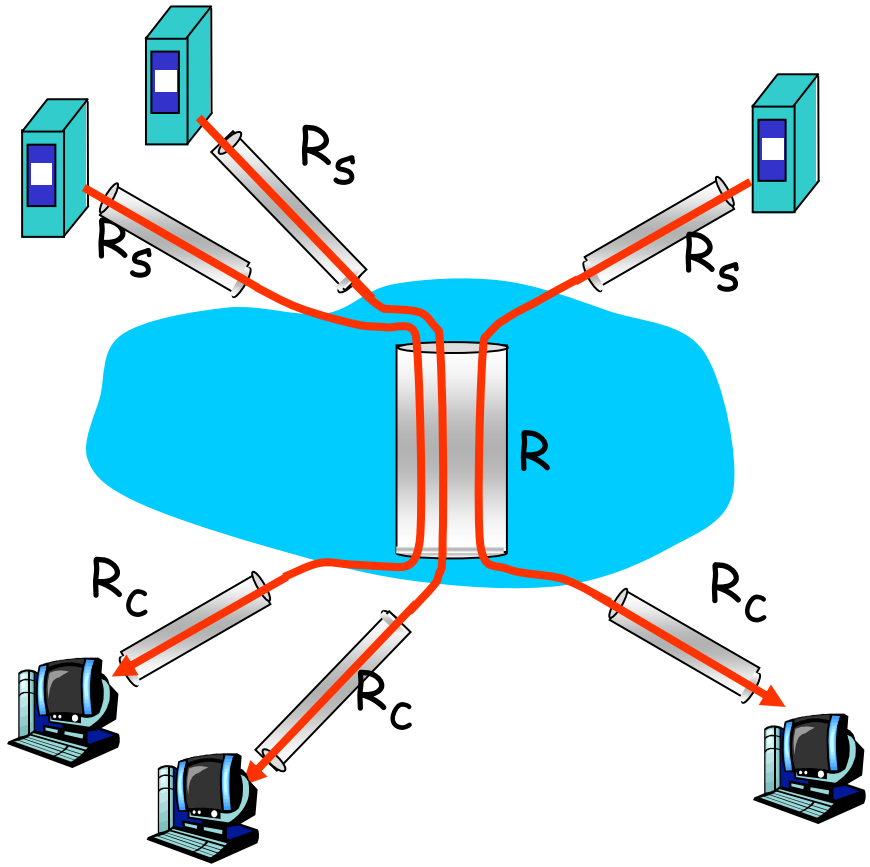


Collo di bottiglia

Collegamento su un percorso punto-punto che vincola un throughput end to end

Throughput: scenario Internet

- throughput end to end per ciascuna connessione: $\min(R_c,R_s,R/10)$
- in pratica: R_c o R_s è spesso nel collo di bottiglia



10 collegamenti (equamente) condivisi
collegamento collo di bottiglia R bit/sec

Introduzione ...

Cos'è Internet?

Ai confini della rete

- sistemi terminali, reti di accesso, collegamenti

Il nucleo della rete

- commutazione di circuito e di pacchetto, struttura della rete

Ritardi, perdite e throughput nelle reti a commutazione di pacchetto

Livelli di protocollo e loro modelli di servizio

Reti sotto attacco: la sicurezza

Livelli di protocollo

- Le reti sono complesse!
- molti "pezzi":
  - host
  - router
  - svariate tipologie di mezzi trasmissivi
  - applicazioni
  - protocolli
  - hardware, software

- Domanda:
- C'è qualche speranza di *organizzare* l'architettura delle reti?

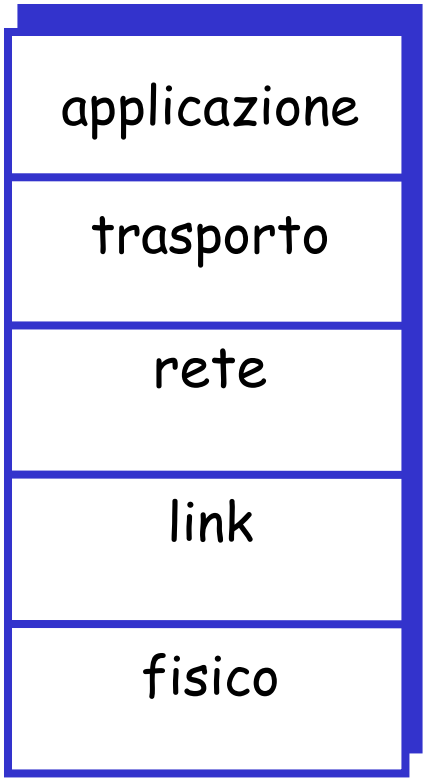
- O almeno la nostra trattazione sulle reti?

Perché la stratificazione?

- Quando si ha a che fare con sistemi complessi:
- Una struttura "esplicita" consente l'identificazione dei vari componenti di un sistema complesso e delle loro inter-relazioni
  - analisi del modello di riferimento a strati
- La modularizzazione facilita la manutenzione e l'aggiornamento di un sistema
  - modifiche implementative al servizio di uno dei livelli risultano trasparenti al resto del sistema
  - es.: modifiche nelle procedure effettuate al gate non condizionano il resto del sistema
- Il modello a strati può essere considerato dannoso?

Pila di protocolli Internet

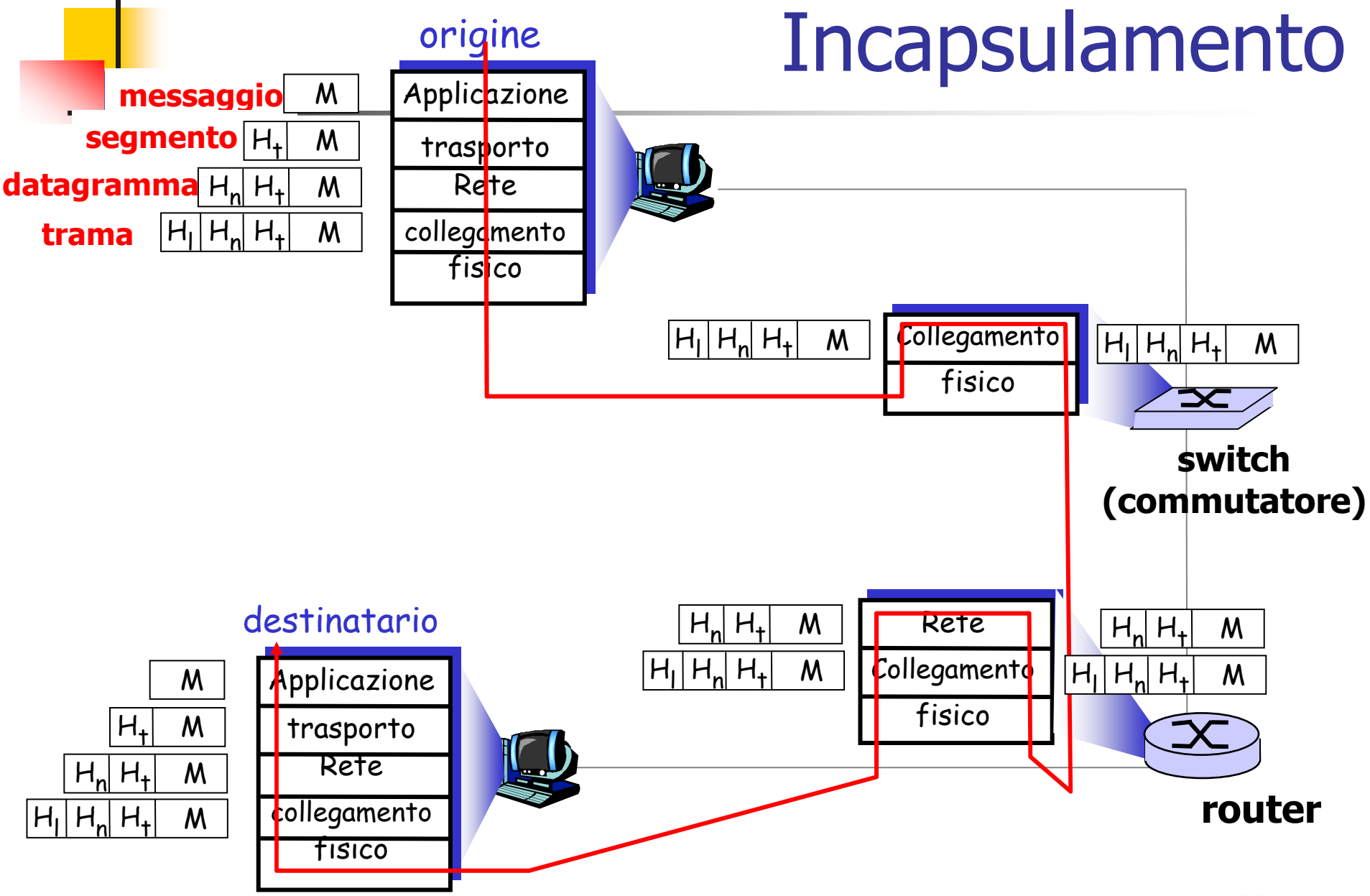
- applicazione: di supporto alle applicazioni di rete
  - FTP, SMTP, HTTP
- trasporto: trasferimento dei messaggi a livello di applicazione tra il modulo client e server di un'applicazione
  - TCP, UDP
- rete: instradamento dei datagrammi dall'origine al destinatario
  - IP, protocolli di instradamento
- link (collegamento): instradamento dei datagrammi attaverso una serie di commutatori di pacchetto
  - PPP, Ethernet
- fisico: trasferimento dei singoli bit



Modello di riferimento ISO/OSI

- presentazione: consente alle applicazioni di interpretare il significato dei dati (es. cifratura, compressione, convenzioni specifiche della macchina)
- *sessione:* sincronizzazione, controllo, recupero dei dati
- La pila Internet è priva di questi due livelli!
  - questi servizi, *se necessario,* possono essere implementati nelle applicazioni
  - sono necessari?

Incapsulamento



Modello di comunicazione

- Quando un sistema vuole scambiare informazioni con un altro sistema, nasce il problema della ***comunicazione***
- La comunicazione tra sistemi racchiude in sé due sotto-problemi:
  1. il linguaggio utilizzato
  2. la modalità di scambio (trasmissione) delle informazioni
- Il primo passo per affrontare la questione è creare un ***modello*** che ne descriva le caratteristiche
- In tale modello, chiameremo:
  - linguaggio: ***comunicazione logica***
  - modalità di scambio: ***comunicazione fisica***

Esempio


comunicazione logica (colloquio)
comunicazione fisica (trasmissione)
canale di comunicazione

Problemi associati alla comunicazione

- Logica
  - linguaggio utilizzato
    - significato dei messaggi
  - regole per lo scambio di informazione
    - modalità instaurazione connessioni
    - algoritmi per instradamento
  - modalità di trasferimento (simplex, half duplex, ...)
  - ...
- Fisica
  - indirizzamento
  - controllo degli errori
  - affidabilità
  - sequenzialità
  - segmentazione
  - multiplazione
  - controllo di flusso
  - modalità di trasmissione del segnale
  - ...

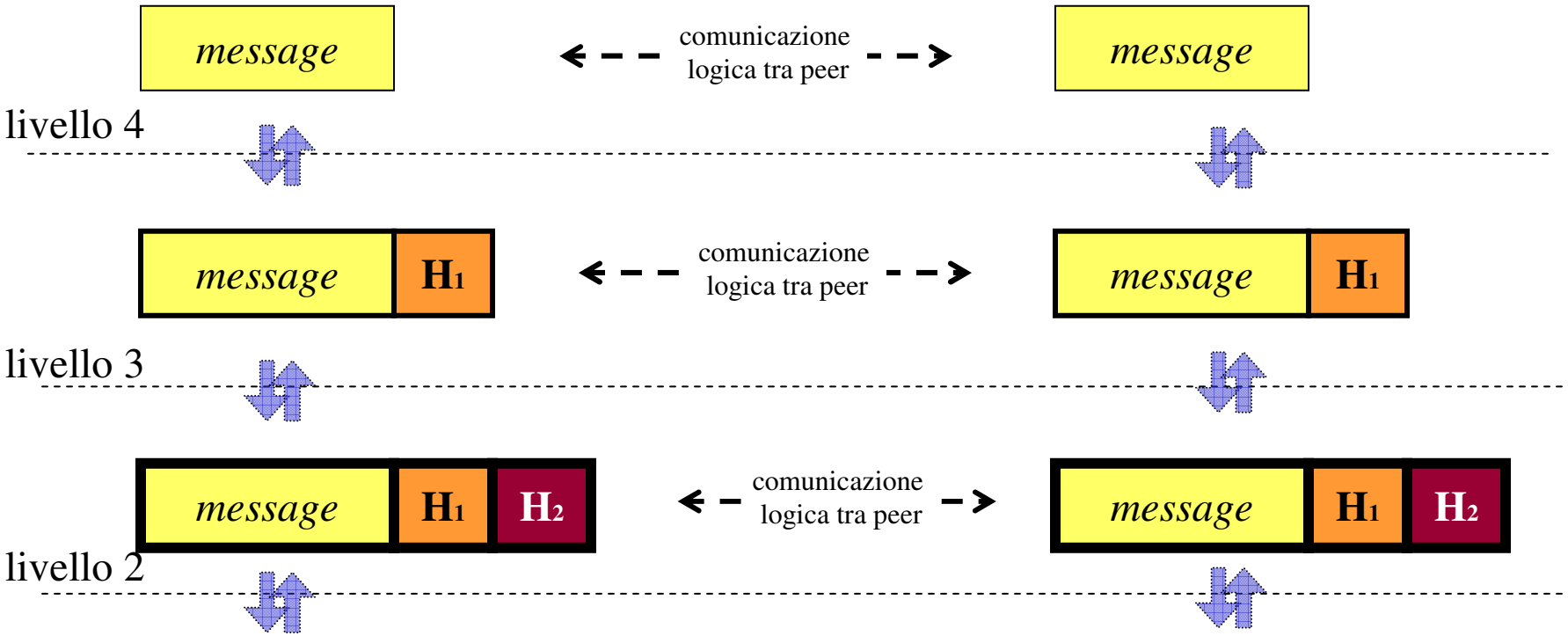
Approccio a livelli

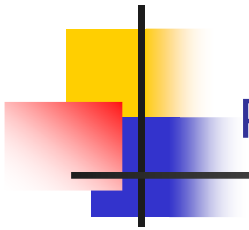
- Approccio "*divide et impera*"
  - il problema della comunicazione (logica e fisica) viene suddiviso in sotto-problemi
  - ciascun sotto-problema viene trattato separatamente
- L'informazione passa attraverso una "catena di montaggio" in cui essa viene trasformata in modo da poter essere spedita
  - ogni passo della catena assolve ad una funzione specifica
    - confezione del prodotto, imballaggio, decisione della destinazione, ...
- Dall'altra parte ci sarà una catena di montaggio *inversa* che restituisce l'informazione
- Tradizionalmente questa catena di montaggio viene rappresentata "in verticale", come una serie di **livelli** (o *layer*, strati)
  - ogni livello assolve ad un compito ben preciso e svolge una serie di funzioni specifiche

Modello a strati

- Ogni livello interagisce solo con i due adiacenti (*comunicazione fisica*):
  - riceve il messaggio dal livello superiore (o inferiore)
  - lo elabora
  - lo passa al livello inferiore (o superiore)
- In genere, nell'elaborazione del messaggio, viene aggiunta dell'informazione
  - l'informazione aggiunta non e' altro che il risultato della funzioni svolte da quel livello
- Il livello N colloquia con il suo omologo (***peer***) di un'altra macchina (*comunicazione logica*)

Esempio

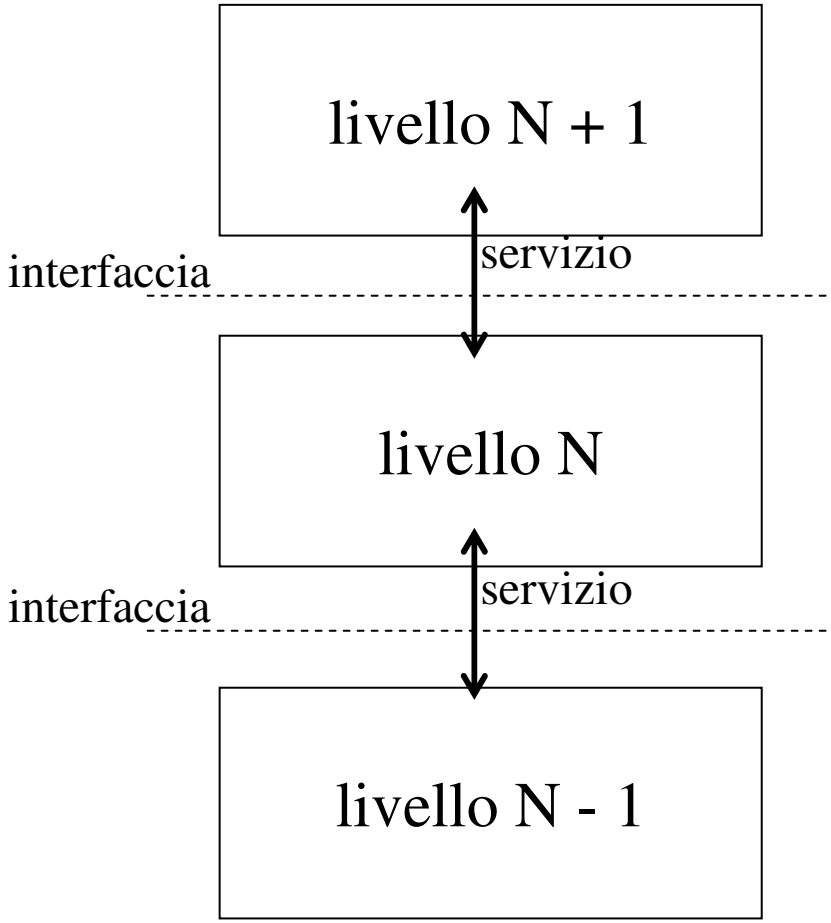

message
comunicazione logica tra peer
message
livello 4
message
H1
comunicazione logica tra peer
message
H1
livello 3
message
H1
H2
comunicazione logica tra peer
message
H1
H2
livello 2
livello 1



Perché il modello a strati?

- Per ogni livello vengono definiti
  - i servizi che esso deve offrire
  - le funzioni che deve svolgere
  - le primitive che deve mettere a disposizione
- **NON** viene definito il **MODO** in cui implementare i servizi / funzioni
  - modularità
  - intercambiabilità
- Rientra nella concezione di "divide et impera" / catena di montaggio
  - esempio: confezionamento di un prodotto
    - alla postazione N della catena, il prodotto viene confezionato (la postazione offre un servizio di confezionamento; la funzione da svolgere sono controllo del riempimento, suddivisione del prodotto in più confezioni, ...)
    - non ha importanza se il confezionamento viene fatto da una macchina o da una persona

Definizioni



- Tra ogni coppia di livelli adiacenti esiste un'***interfaccia***
- L'interfaccia definisce i ***servizi*** offerti dal livello sottostante a quello superiore
- Ogni livello può offrire più di un servizio al livello superiore
- Per espletare il servizio, ogni livello compie una serie di ***funzioni***
- I servizi vengono fruiti attraverso ***primitive***

Servizi, funzioni e primitive

- Gli elementi attivi in ogni livello del sistema vengono detti ***entità***
- Un ***servizio*** è una prestazione fornita dall'entità di livello inferiore ad una entità di livello superiore
- Le ***funzioni*** sono un'insieme di attività (elaborazione, analisi, aggiunte) che nell'insieme creano il servizio
  - Per poter espletare un servizio, l'entità svolge una serie di funzioni
- Le ***primitive*** sono delle comunicazioni tra entità per poter usufruire del servizio offerto (richiesta del servizio e ricezione di informazioni sul servizio)
  - sono caratterizzate da parametri tra cui: informazione da trasferire, indicazione del destinatario, caratteristiche del servizio richiesto, ...

- In definitiva:
  - attraverso le primitive, un livello richiede al livello sottostante un servizio; il servizio viene soddisfatto attraverso lo svolgimento di funzioni

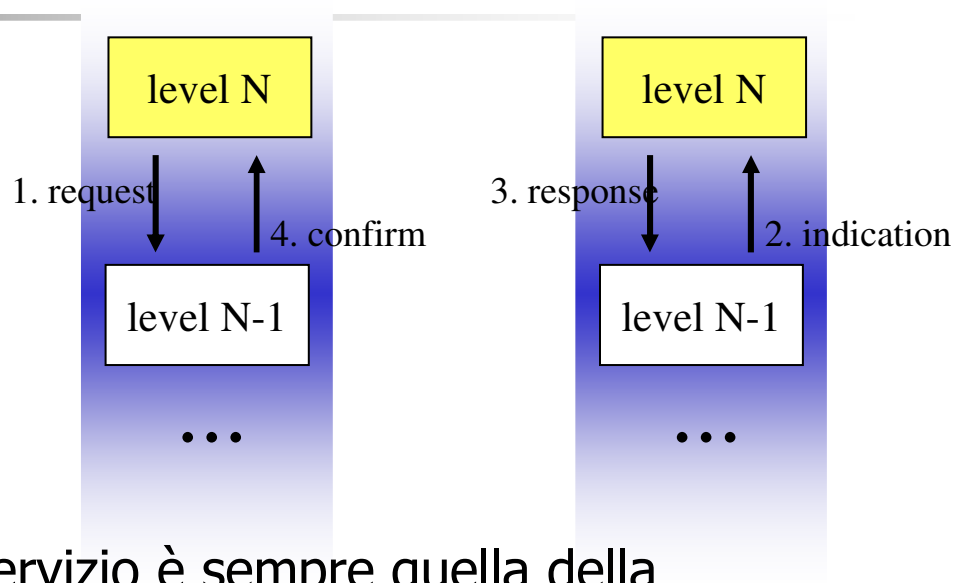
Primitive

Request: richiesta di un servizio

Indication: indicazione di evento

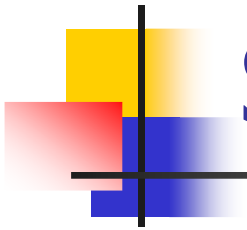
Response: risposta all'indicazione

Confirm: conferma della richiesta



- La finalità di una richiesta di un servizio è sempre quella della comunicazione, ovvero se un'entità fa una richiesta di servizio è perché vuole comunicare con la sua entità pari
- Segue che, per ogni Request, esiste un'Indication all'entità pari
- Inoltre, se viene richiesto il riscontro, ad una coppia Request – Indication corrisponde una coppia Response – Confirm

Nota: *le primitive hanno carattere locale ed è il linguaggio utilizzato dai diversi livelli per comunicare tra loro*



Servizi

- Esistono due **categorie** di servizi
  - connection oriented
    - in questa categoria ricadono i servizi punto-punto
      - garantiscono la consegna sequenziale
  - connectionless
    - in questa categoria ricadono i servizi che non si preoccupano di instaurare una connessione, ma prevedono il semplice "passaggio" dell'informazione
      - su modello del sistema postale
- Per ciascuna categoria è possibile inoltre associare una "qualità del servizio" (Quality of Service, QoS)
  - affidabile, non affidabile, ritardo, perdite, errori ...

Esempi di servizi

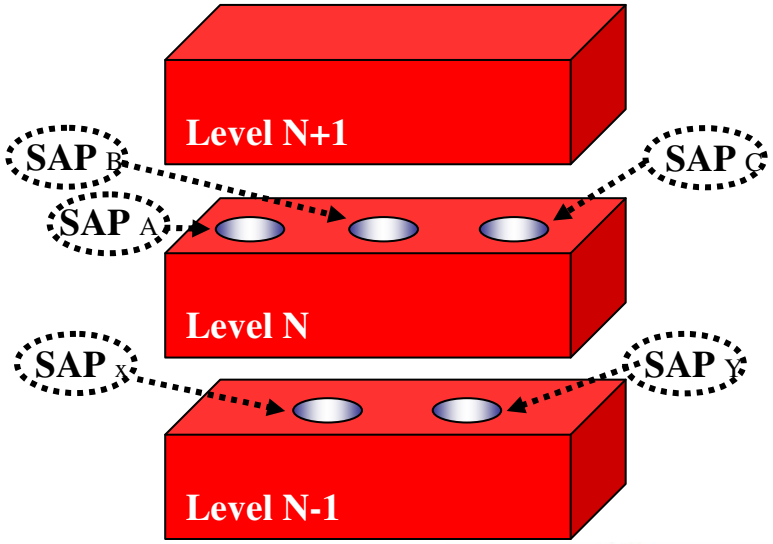
QoS		Categorie di servizio	
		Connection oriented	***Connectionless***
	Affidabili	Visione di pagine web	Posta elettronica con ricevuta
	Non affidabili	Applicazione multimediale	Posta elettronica

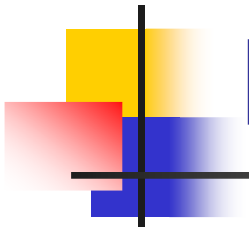
- Questi sono esempi di alto livello per capire i concetti base
- Ogni livello offre servizi che rientrano in una delle due categorie di servizio e con una determinata QoS, questo indipendentemente dai servizi offerti dal livello inferiore
  - Per questo motivo può capitare che il livello N offra al livello N+1 servizi connection oriented affidabili, ma che si veda offrire dal livello N-1 solo servizi connectionless non affidabili e a sua volta il livello N-2 offre a livello N-1 servizi connectionless affidabili

Servizi

- Attenzione: ogni livello può offrire al livello superiore una serie di servizi
- I servizi offerti possono rientrare nelle due categorie e posso essere affidabili e non.
- La scelta di uno o l'altro servizio viene fatta accedendo al livello attraverso un punto particolare a seconda del servizio richiesto ➔ **SAP** (Service Access Point)

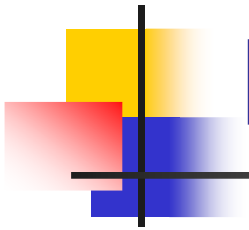
- Esempio:
  - servizio A: connection oriented affidabile
  - servizio B: connection oriented non affidabile
  - servizio C: connectionless non affidabile
  - servizio X: connection oriented affidabile
  - servizio Y: connectionless affidabile





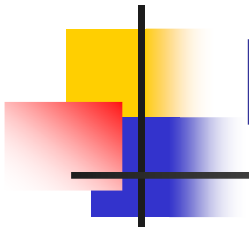
Funzioni

- Per fornire il servizio, ogni livello svolge una serie di funzioni
- Alcuni tipi di funzioni posso essere:
  - Instaurazione/terminazione delle connessioni
  - Controllo d'errore e controllo di flusso
  - Riordino trame
  - Multiplazione
  - Segmentazione
  - Instradamento
  - Indirizzamento
  - . . .



Definizioni

- Finora è stata fatta una panoramica dell'interazione tra livelli
  ➔ ***comunicazione fisica***
- Ricordiamo che lo scopo finale è la comunicazione tra entità pari, ovvero dello stesso livello su due macchine differenti
  ➔ ***comunicazione logica***
- Le problematiche associate alla *comunicazione fisica* vengono risolte con il modello a strati, definendo le primitive, i servizi e le funzioni di ciascun livello
- Le problematiche associate alla *comunicazione logica* vengono risolte attraverso la definizione di ***protocolli***



Definizioni

Protocollo

insieme di regole che sovraintendono al colloquio tra entità dello stesso livello

- formato dei messaggi, informazioni di servizio, algoritmi di trasferimento, etc.

- ogni livello ha il suo protocollo specifico che è comprensibile solo dalle entità dello stesso livello
- le entità di livello diverso trattano il contenuto come fosse una scatola chiusa

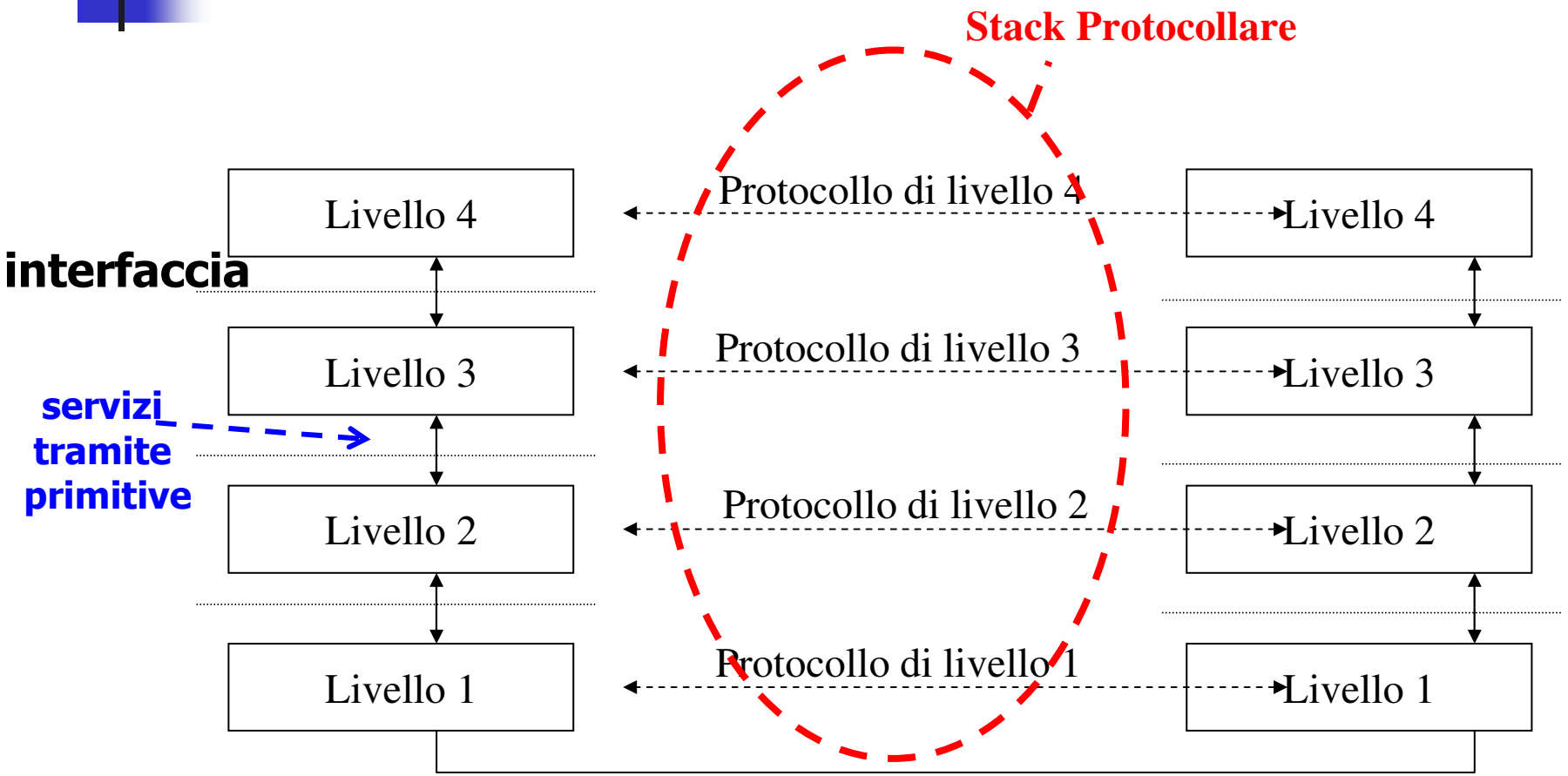
Stack protocollare

insieme dei protocolli di ciascun livello

Architettura di rete

l'insieme dei livelli e dei rispettivi protocolli

Esempio: architettura di rete



Nota: la comunicazione tra livelli avviene tramite primitive mentre la comunicazione tra entità pari (dello stesso livello) di sistemi diversi avviene tramite protocolli

Esempio: architettura a 4 livelli

- Per ogni livello vengono definiti le funzionalità che deve svolgere, ovvero i servizi che deve offrire ai livelli adiacenti
- Vengono specificate delle primitive che hanno significato a livello locale, ovvero vengono comprese solo dai livelli adiacenti che le usano
- Per ogni livello viene definito il protocollo che usa: tale protocollo risulta incomprensibile ai livelli adiacenti, ma comprensibile al livello corrispondente (comunicazione logica)
- esempio: telescrivente
  - i due livelli applicativi sanno che l'informazione sono dei caratteri (ASCII)
  - al livello di trasporto viene richiesto di instaurare una connessione e trasferire i dati (questo tramite primitive); oltre a tale richiesta viene passato il messaggio
  - il livello di trasporto vede solo dei bit, ma non ne comprende il significato: deve solo trasportarli

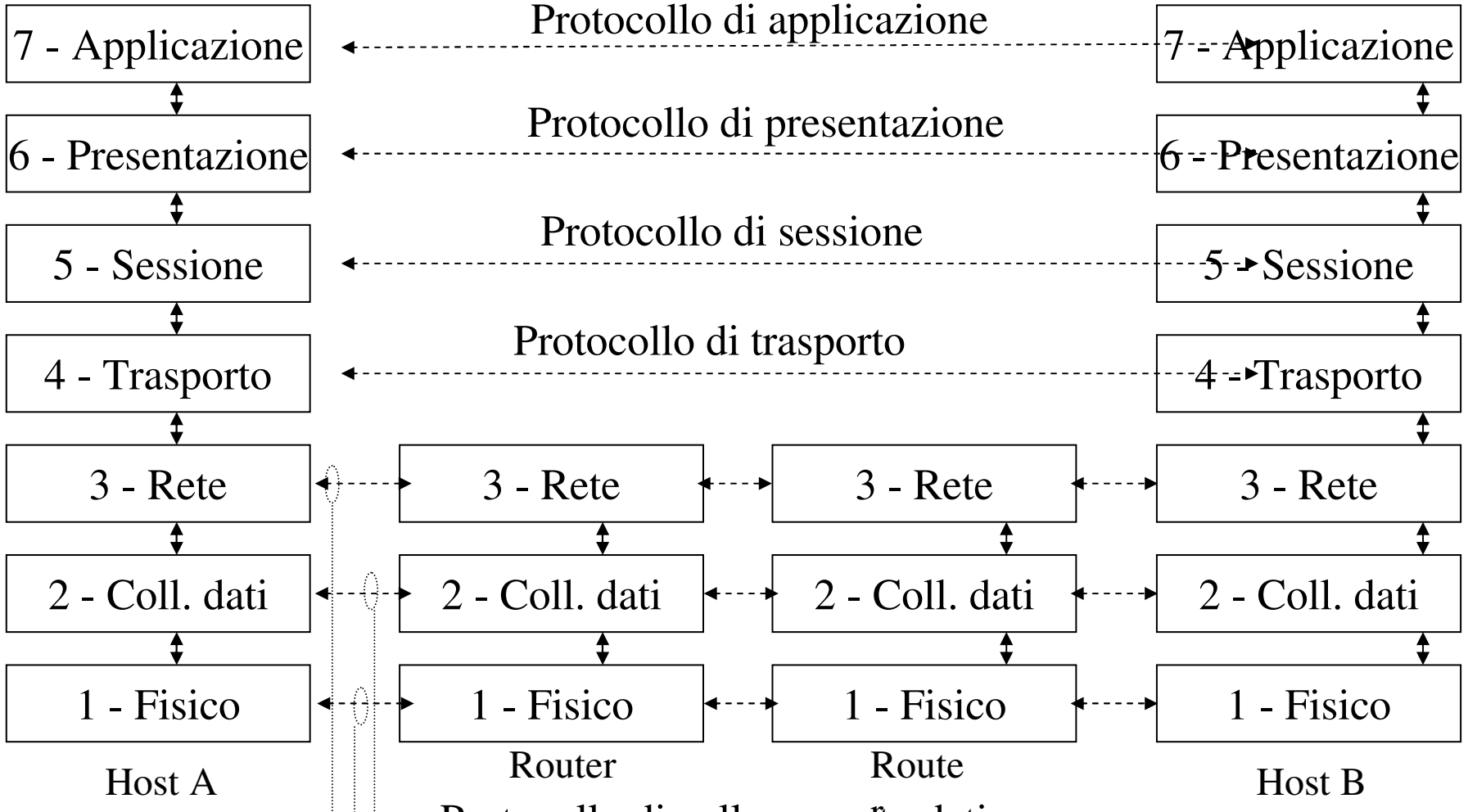
Esempio di architettura di rete: il modello di riferimento ISO-OSI

- È stato il primo passo nella definizione di un'architettura di rete completa e aperta (non proprietaria)
  - ISO: International Standard Organization
  - OSI: Open System Interconnection
  - modello per l'interconnessione dei sistemi aperti, ovvero dei sistemi che sono aperti alla comunicazione con altri sistemi.
- Essendo un primo passo nella definizione dell'architettura, è un modello che definisce ***funzionalità*** raggruppate in livelli, ma non ancora in modo formale protocolli e servizi da usare nei vari livelli
  - non è dunque un'architettura di rete vera e propria
- Standardizzato nel 1983
  - Modello teorico sviluppato troppo tardi
    - alla pubblicazione di OSI Internet era già una realtà
    - utilizzato come modello di riferimento

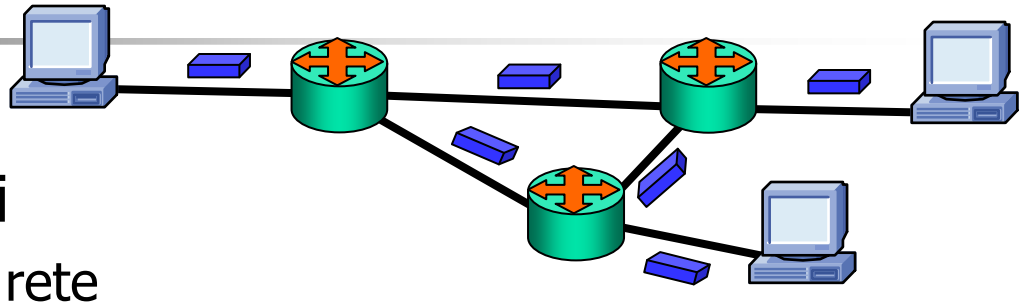
Modello OSI: principi

- Un livello deve essere creato per ogni grado di astrazione
- Ogni livello deve eseguire funzioni ben definite
- Le interfacce tra i livelli devono essere definite in modo da minimizzare l'informazione scambiata
- Il numero di livelli deve essere:
  - sufficientemente grande in modo che le stesse funzioni non siano separate in più livelli
  - sufficientemente piccolo in modo che l'architettura non risulti con funzionalità ridondate

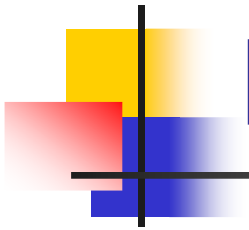
Stack OSI



Modello OSI



- Strutturato in sette livelli
  - primi tre dipendenti dalla rete
  - ultimi tre dipendenti dall'applicazione
  - quarto livello isola ambiente rete da ambiente applicazione
- Sono stati definiti 2 sistemi distinti:
  - end system (host): è l'end user coinvolto nella comunicazione
  - intermediate system (router): è un elemento il cui compito è solo il trasporto del messaggio
- Poiché le funzionalità del router sono quelle di trasporto indipendentemente dal contenuto del messaggio, non è necessario che siano implementati tutti i livelli



Livello fisico

- Gestisce la trasmissione del segnale su canale fisico
- Funzioni
  - trasferimento di un flusso seriale di bit
  - attivazione, disattivazione e controllo della connessione fisica
- Protocolli
  - specificano le caratteristiche elettriche, meccaniche e procedurali
    - ad esempio: trasmissione on-off o antipodale, significato dell'ordine dei bit, formato della flag, …;

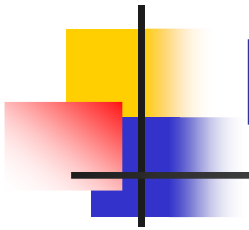
Livello di data link

- Fornisce un canale numerico di comunicazione il più possibile affidabile
  - trasferimento di unità logiche di bit (trame) su un collegamento
- Funzioni:
  - gestione collegamento
  - framing (divisione delle trame)
  - controllo errori
  - controllo di flusso
- Protocolli
  - definiscono il formato della trama
  - definiscono i messaggi di feedback per il controllo di flusso
  - definiscono gli algoritmi per la gestione trasmissione

da questo livello in poi i servizi offerti possono sempre appartenere alle due categorie (**connectionless** e **connection oriented**) per cui questo aspetto non verrà più citato

Livello di rete

- E' responsabile del trasferimento di informazioni tra nodi, indipendentemente dal tipo di collegamento
- Funzioni:
  - instradamento
  - internetworking
- Protocolli
  - definizione del formato dei pacchetti
  - definizione dei messaggi per lo scambio di informazioni
  - definizione degli algoritmi per l'instradamento (shortest path, optimal routing)



Livello di trasporto

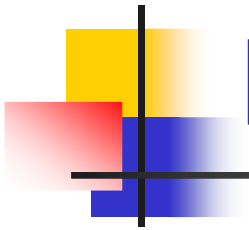
- Fornisce un canale di trasporto ideale e privo di errori tra due utenti, indipendentemente dalla rete
- Funzioni
  - recupero degli errori
  - multiplazione / demultiplazione
  - riordino dei pacchetti
  - controllo della congestione
- Protocolli
  - definizione del formato dei pacchetti
  - definizione dei messaggi per lo scambio di informazioni
  - definizione degli algoritmi per il controllo della congestione

Livello di sessione

- Consente a due applicazioni di sincronizzarsi e gestire lo scambio dei dati
- Funzioni:
  - instaurazione e rilascio di una *connessione di sessione*
  - scambio di dati normali e di dati con priorità
  - gestione del dialogo tra entità comunicanti mediante token
  - sincronizzazione e strutturazione del dialogo
  - gestione delle eccezioni
- Protocolli
  - definizione del formato dei pacchetti
  - definizione dei messaggi per lo scambio di informazioni
  - definizione degli algoritmi per il controllo della sessione

Livello di presentazione

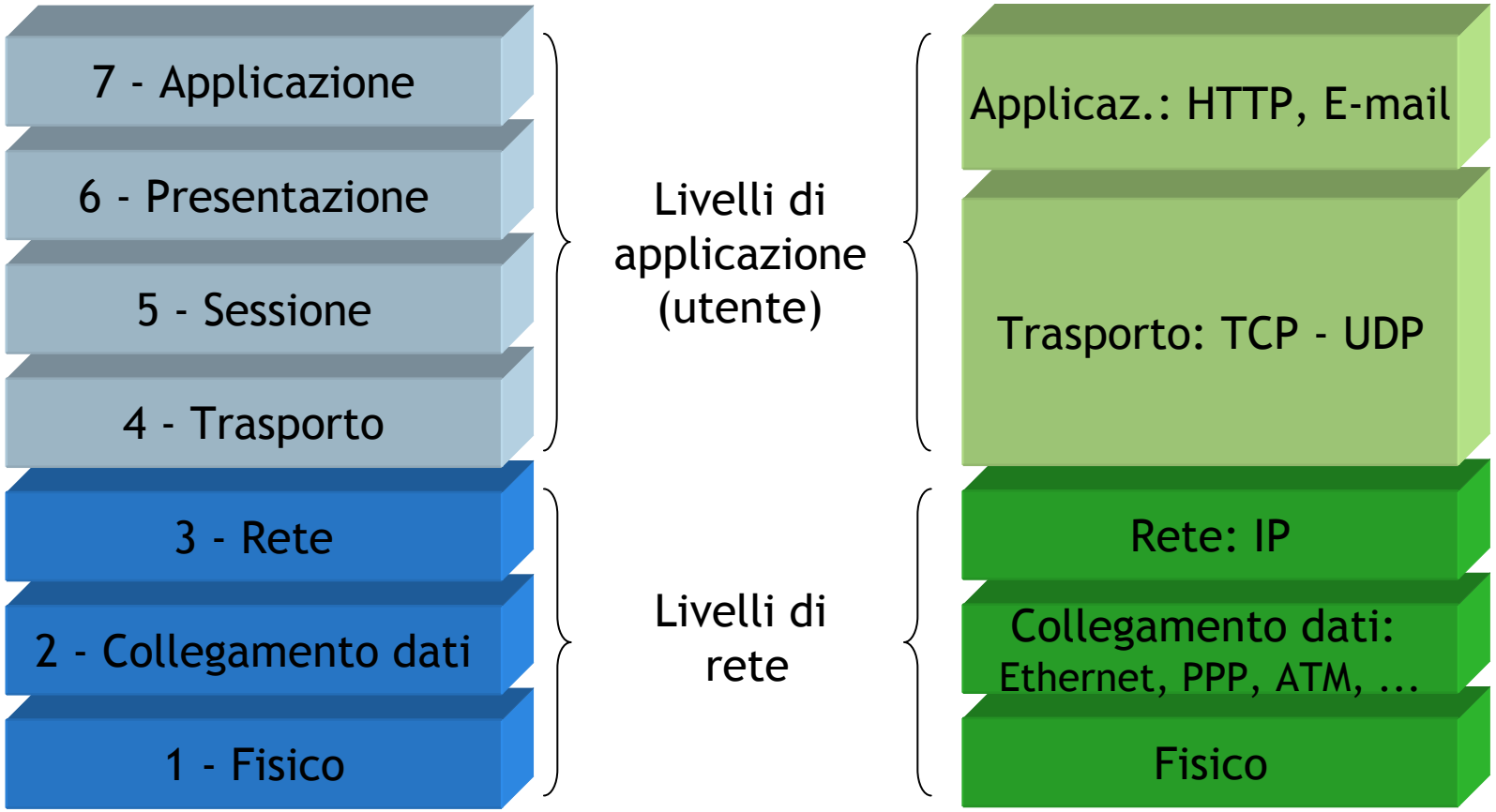
- Si occupa dei problemi relativi alla rappresentazione dei dati
  - sintassi dell'informazione
- Funzioni
  - conversione dei dati dal formato di trasmissione ad un formato utile all'applicazione
    - codifica e decodifica
    - compressione dei dati
    - crittografia
- Protocolli
  - definizione del formato dei pacchetti
  - definizione strutture dati complesse
  - definizione dei messaggi per lo scambio di informazioni
  - definizione degli algoritmi per codifica/decodifica, compressione, crittografia, ...

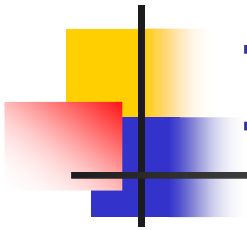


Livello delle applicazioni

- Fornisce i servizi (applicazioni) all'utente
- Fra queste:
  - login remoto
  - file transfer
  - servizi WWW
  - e-mail
  - ...

Stack OSI ... Stack TCP/IP





Introduzione ...

Cos'è Internet?

Ai confini della rete

- sistemi terminali, reti di accesso, collegamenti

Il nucleo della rete

- commutazione di circuito e di pacchetto, struttura della rete

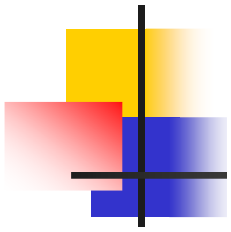
Ritardi, perdite e throughput nelle reti a commutazione di pacchetto

Livelli di protocollo e loro modelli di servizio

Reti sotto attacco: la sicurezza

Sicurezza di rete

- Il campo della sicurezza di rete si occupa di:
  - malintenzionati che attaccano le reti di calcolatori
  - come difendere le reti dagli attacchi
  - come progettare architetture immuni da attacchi
- Internet non fu inizialmente progettato per la sicurezza
  - *Visione originaria:* "un gruppo di utenti che si fidavano l'uno dell'altro collegati a una rete trasparente" ☺
  - I progettisti del protocollo Internet stanno recuperando
  - Un occhio alla sicurezza in tutti i livelli



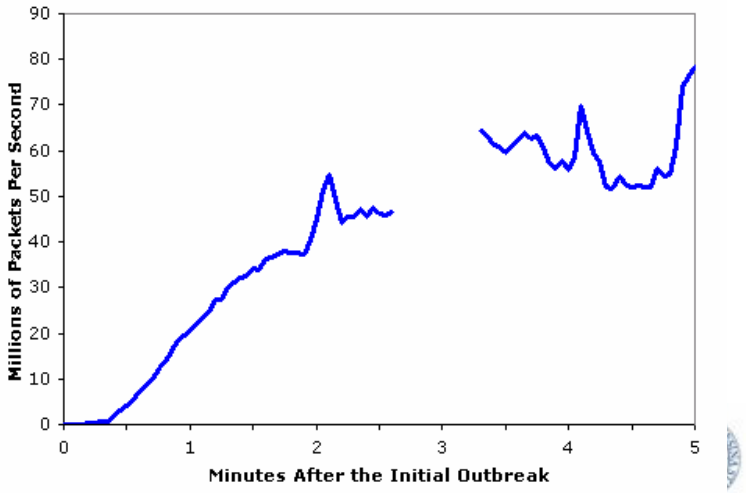
I malintenzionati installano malware negli host attraverso Internet

- Il malware può raggiungere gli host attraverso virus, worm, o cavalli di Troia.
- Malware di spionaggio può registrare quanto viene digitato, i siti visitati e informazioni di upload.
- Gli host infettati possono essere "arruolati" in botnet, e usati per lo spamming e per gli attacchi di DDoS.
- Il malware è spesso auto-replicante: da un host infettato può passare ad altri host

I malintenzionati installano malware negli host attraverso Internet

- Cavalli di Troia
  - Parte nascosta di un software utile
  - Oggi si trova spesso su alcune pagine web (Active-X, plugin)…
- Virus
  - L'infezione proviene da un oggetto ricevuto (attachment di e-mail), e mandato in esecuzione
  - Auto-replicante: si propaga da solo ad altri host e utenti
- Worm:
  - L'infezione proviene da un oggetto passivamente ricevuto che si auto-esegue
  - Auto-replicante: si propaga da solo ad altri host e utenti

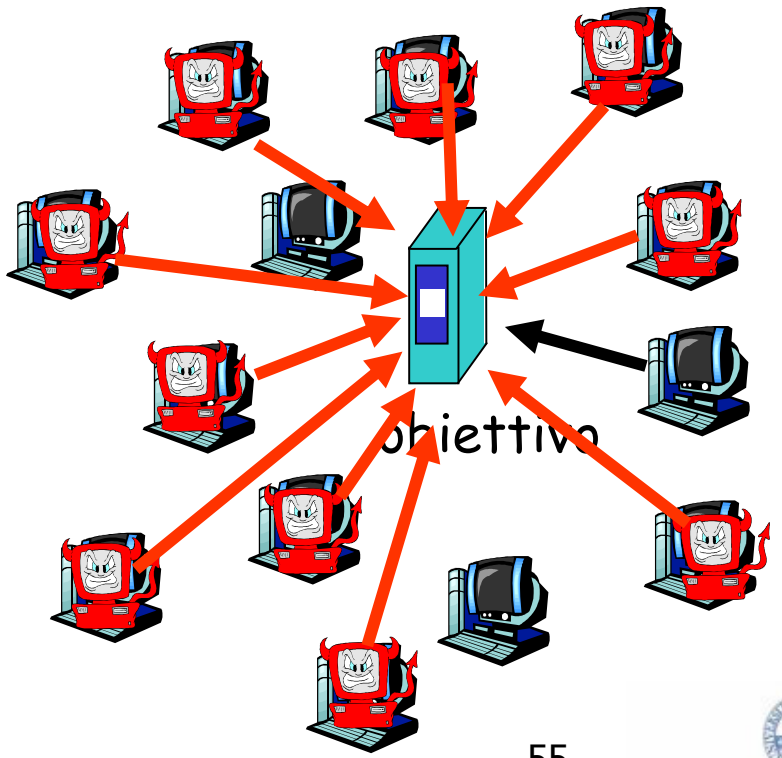
Worm Sapphire : scans/sec aggregati
nei primi 5 minuti di diffusione (CAIDA, UWisc data)



I malintenzionati attaccano server e infrastrutture di rete

- Negazione di servizio (DoS): gli attaccanti fanno sì che le risorse (server, ampiezza di banda) non siano più disponibili al traffico legittimo sovraccaricandole di traffico artefatto

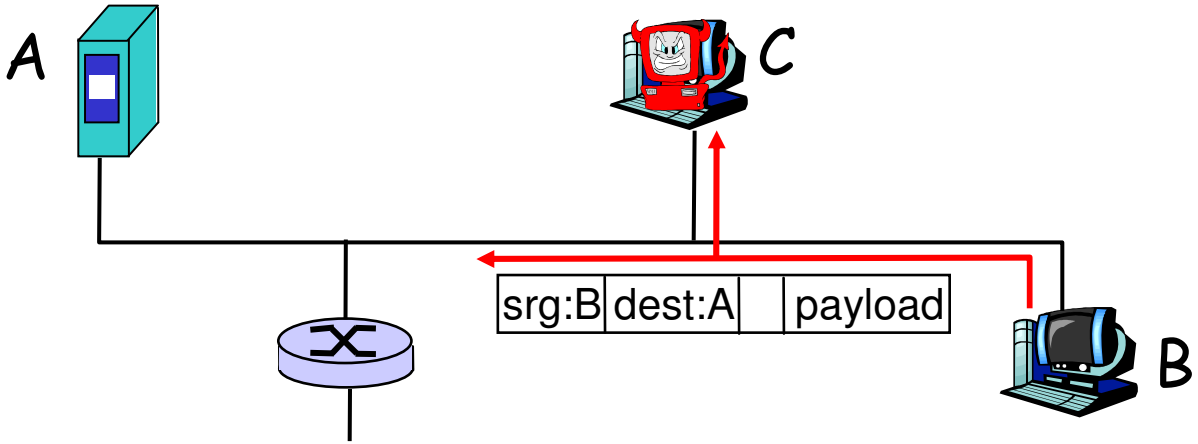
- Selezione dell'obiettivo
- Irruzione negli host attraverso la rete
- Invio di pacchetti verso un obiettivo da parte degli host compromessi



I malintenzionati analizzano i pacchetti

Analisi dei pacchetti (packet sniffing):

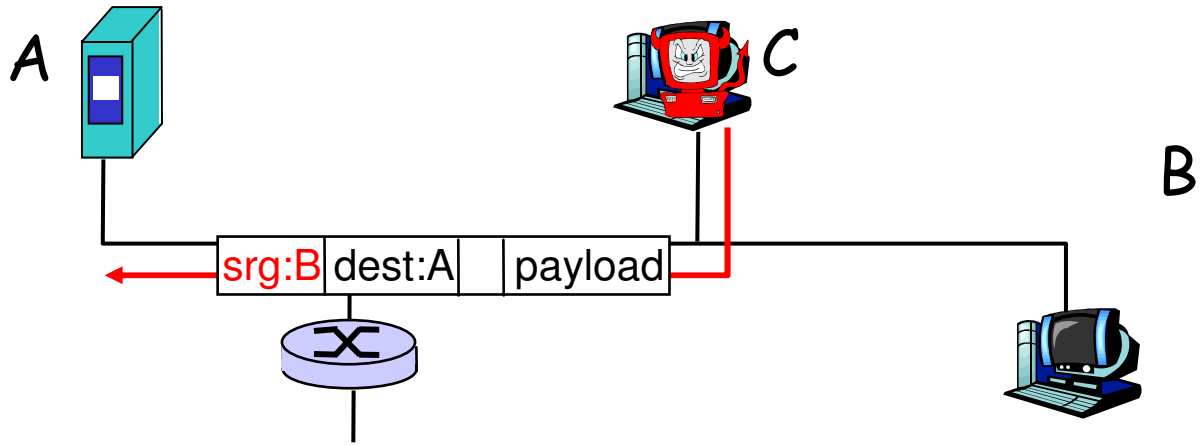
- media broadcast (Ethernet condivisa, wireless)
- un'interfaccia di rete legge/registra tutti i pacchetti (password comprese!) che l'attraversano



- Il software usato per i laboratori è un packet-sniffer (gratis!)

I malintenzionati usano indirizzi sorgente falsi

- *IP spoofing:* invio di pacchetti con un indirizzo sorgente falso



I malintenzionati registrano e riproducono

- *record-and-playback*: "sniffano" dati sensibili (password, ad esempio), per poi utilizzarli in un secondo tempo